

(19)日本国特許庁 (J P)

(12) 公開特許公報 (A)

(11)特許出願公開番号

特開平7-84921

(43)公開日 平成7年(1995)3月31日

(51) Int.Cl. <sup>a</sup>	識別記号	序内整理番号	F I	技術表示箇所
G 0 6 F 13/00	3 5 5	7368-5B		
H 0 4 L 12/40				
	7341-5K		H 0 4 L 11/ 00	3 2 0

審査請求 未請求 請求項の数4 FD (全14頁)

(21)出願番号 特願平5-183145

(22)出願日 平成5年(1993)6月30日

(71)出願人 000006747

株式会社リコー

東京都大田区中馬込1丁目3番6号

(72)発明者 森田 岳彦

東京都大田区中馬込1丁目3番6号 株式  
会社リコー内

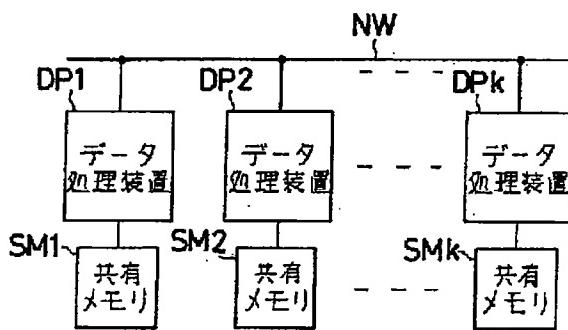
(74)代理人 弁理士 紋田 誠

(54)【発明の名称】 ネットワークシステム

(57)【要約】

【目的】 データやメモリの共有を円滑にかつ柔軟に実現できるネットワークシステムを提供することを目的としている。

【構成】 異なるノード間でメモリを共有できるので、ネットワークシステム全体でのデータの共有を、容易に実現することができる。また、仮想記憶管理機構を備えている場合には、共有メモリ空間を仮想記憶管理機構を用いて直接アクセスすることができるので、データのアクセス効率がよく、また、共有メモリを高速にアクセスすることができる。



**【特許請求の範囲】**

**【請求項1】** 複数ノードを接続したネットワークシステムにおいて、おのおののノードに、他のノードから共用できるメモリオブジェクトを保存する共有メモリ領域を設けたことを特徴とするネットワークシステム。

**【請求項2】** 前記共有メモリ領域は、そのネットワークシステム内に定義された仮想的な線形メモリ領域の一部を構成することを特徴とする請求項1記載のネットワークシステム。

**【請求項3】** 前記共有メモリ領域に保存される前記メモリオブジェクトには、ユニークな識別子が付加され、各ノードからは、その識別子を用いてアクセスできることを特徴とする請求項1記載のネットワークシステム。

**【請求項4】** おのおのが仮想記憶管理機構を備えた複数ノードを接続したネットワークシステムにおいて、おのおののノードに、他のノードから共用できるメモリオブジェクトを保存する共有メモリ領域を設けるとともに、その共有メモリ領域には、上記仮想記憶メモリ管理機構に対応した仮想記憶管理アドレスが付されていることを特徴とするネットワークシステム。

**【発明の詳細な説明】****【0001】**

**【産業上の利用分野】** 本発明は、複数ノードを接続したネットワークシステムに関する。

**【0002】**

**【従来の技術】** 例えば、ローカルエリアネットワークなどのネットワークを介して、パーソナルコンピュータやワークステーションなどの複数のノード（データ処理装置）を接続し、これらの複数のノード間でデータオブジェクト（ファイル、実行可能なデータなど）を共有する方法としては、従来、次のようなものが用いられている。

**【0003】** 例えば、2つのノード間で1対1の通信を実行することで、必要なデータオブジェクトをやりとりする方法や、データ受信用のポスト（メール・ボックスなど）を設け、データ送信側からこのポストに対してデータオブジェクトを送信する通信方法などがある。この後者の方法では、1対多の通信が実現できる。また、1対多の通信方法としては、ローカルエリアネットワークのブロードキャスト機能を用いる方法もある。

**【0004】**

**【発明が解決しようとする課題】** しかしながら、このような従来では、データの共有方法が、主としてデータの送受信という形式をとるために、相手先ノードの特定のための処理が必要であったり、また、目的以外の相手先ノードに対しても、データが送信されるという不都合が生じる。

**【0005】** また、ネットワークシステム全体でデータを共有してシステムを円滑に運営する場合、相手先ノードを接続したネットワークシステムにおいて、おのおののノードに、他のノードから共用できるメモリオブジェクトを保存する共有メモリ領域を設けたことを特徴とするネットワークシステム。

ドを指定するような指向性のある通信ではなく、単に、ネットワークシステム全体でメモリやデータを共有したいという応用には、上述した従来方法は適さない。

**【0006】** 本発明は、かかる実情に鑑みてなされたものであり、データやメモリの共有を円滑にかつ柔軟に実現できるネットワークシステムを提供することを目的としている。

**【0007】**

**【課題を解決するための手段】** 本発明は、複数ノードを接続したネットワークシステムにおいて、おのおののノードに、他のノードから共用できるメモリオブジェクトを保存する共有メモリ領域を設けたものである。また、前記共有メモリ領域は、そのネットワークシステム内で定義した仮想的な線形メモリ領域の一部を構成する。また、前記共有メモリ領域に保存される前記メモリオブジェクトには、ユニークな識別子を付加し、各ノードからは、その識別子を用いてアクセスできるようにするといふ。

**【0008】** また、おのおのが仮想記憶管理機構を備えた複数ノードを接続したネットワークシステムにおいて、おのおののノードに、他のノードから共用できるメモリオブジェクトを保存する共有メモリ領域を設けるとともに、その共有メモリ領域には、上記仮想記憶メモリ管理機構に対応した仮想記憶管理アドレスを付加するようにしたものである。

**【0009】**

**【作用】** したがって、異なるノード間でメモリを共有できるので、ネットワークシステム全体でのデータの共有を、容易に実現することができる。また、仮想記憶管理機構を備えている場合には、共有メモリ空間を仮想記憶管理機構を用いて直接アクセスすることができるので、データのアクセス効率がよく、また、共有メモリを高速にアクセスすることができる。

**【0010】**

**【実施例】** 以下、添付図面を参照しながら、本発明の実施例を詳細に説明する。

**【0011】** 図1は、本発明の一実施例にかかるネットワークシステムを示している。

**【0012】** 同図において、ローカルエリアネットワークなどのネットワークNWを介して、パーソナルコンピュータやワークステーションなどの複数のデータ処理装置（ノード）D P 1～D P kが接続されている。また、おのおののデータ処理装置D P 1～D P kには、それぞれ共有メモリSM 1～SM kが設けられている。

**【0013】** ここで、ネットワークNWは、1対1の双方向通信機能、宛先を指定しないブロードキャスト機能などのデータ通信機能を備えている。また、おのおののデータ処理装置D P 1～D P kには、ネットワーク上で指定するためのノードアドレスが設定されている。また、おのおののデータ処理装置D P a～D P kの間のデ

ータのやりとりは、サーバ／クライアントモデルによって実現されている。

【0014】図2は、データ処理装置DP(DP1～DPk)のシステム構成の一例を示している。なお、同図では、本発明の要部に関する部分のみを示しており、他の部分については、説明を省略する。

【0015】同図において、タスク群1は、このデータ処理装置DPで実行されるユーザプロセスを構成するものであり、メモリ2は、タスク群1の実行時に用いられるメモリ領域であり、メモリ管理制御部3は、タスク群1またはリモートメモリ管理制御部4が、メモリ2に対してデータアクセスする動作を管理するためのものである。

【0016】リモートメモリ管理制御部(RMMS)4は、メモリ2に形成する共有メモリSM(SM1～SMk)のデータアクセスのためものであり、自ノードの共有メモリSMに他ノードからアクセスするためのサーバ(RMMS／サーバ)機能と、他ノードの共有メモリSMにアクセスするためのクライアント(RMMS／クライアント)機能を備えている。

【0017】ネットワーク制御部5は、このデータ処理装置DPをネットワークNWに接続するとともに、ネットワークNWの通信機能を用いて、他のノードとの間でデータをやりとりするためのものである。

【0018】メモリ2の領域は、図3に示すように、共有メモリSMを構成するための共有メモリ領域と、自ノード内のメモリオブジェクトなどを記憶するためのローカルメモリ領域に分割されている。

【0019】共有メモリ領域には、図4(a)に示すように、複数のデータオブジェクトが保存されるとともに、同図(b)に示すように、おのおののデータオブジェクトをアクセスするための共有メモリ管理情報が保存される。この共有メモリ管理情報は、同図(c)に示すように、データオブジェクトを保存したアドレス、データオブジェクトのサイズ、このデータオブジェクトをリンク(後述)しているノードの一覧からなるリンク情報、このデータオブジェクトをリンクしているタスクの数をあらわすリンクカウンタ、このデータオブジェクトがロックされているかどうかをあらわすロック情報、および、このデータオブジェクトに対するアクセス制限などをあらわす排他モード情報からなる。

【0020】ローカルメモリ領域は、図5(a)に示すように、さらに、ノードのシステムで利用されるシステム領域、共有メモリのアクセスに関する管理情報を記憶するための管理情報領域、および、ユーザタスク(プロセス)で利用されるユーザ領域に分割されている。

【0021】管理情報領域には、同図(b)に示したように、リンクした共有メモリをアクセスするときに参照するリンク制御情報、および、有効な共有メモリを保持するノードアドレスの一覧からなる共有メモリ一覧情報

が記憶されている。

【0022】リンク制御情報は、同図(c)に示したように、複数の共有メモリ参照情報からなり、おのおのの共有メモリ参照情報は、そのリンクした共有メモリを保持するノードのノードアドレスと、その共有メモリを指定するための参照情報(例えば、アドレスやハンドル(アドレスを指定するためのポインタ))からなる。

【0023】以上の構成で、例えば、データ処理装置DPaが共有メモリ2'を作成し、このデータ処理装置DPaの共有メモリ2'をデータ処理装置DPbのタスクがアクセスする場合を、図6に示す。この場合、共有メモリ2'のアクセス制御動作は、データ処理装置DPaのリモートデータ管理制御部4のサーバ機能と、データ処理装置DPbのリモートデータ管理制御部4のクライアント機能の間で行われる。

【0024】まず、データ処理装置DPaの共有メモリ作成タスク1aは、リモートメモリ管理制御機能呼び出しによって、共有メモリの作成を要求し、これにより、リモートメモリ管理制御部4は、メモリ2の共有メモリ領域に、物理的なメモリオブジェクトを共有メモリ2'として確保する。そして、このときのメモリオブジェクトの実アドレスや、サイズなどに基づいて、共有メモリ管理情報を作成して、保存する。それとともに、その共有メモリ2'をアクセスするために必要な情報(アドレスやハンドル)を、共有メモリ作成タスク1aに返す。

【0025】それとともに、データ処理装置DPaのリモートメモリ管理制御部4は、共有メモリ2'を作成した旨を、他のデータ処理装置DPb～DPkに対して通知する。これにより、データ処理装置DPb～DPkのリモートメモリ管理制御部4は、このデータ処理装置DPaの共有メモリ2'に関して、共有メモリ一覧情報を更新するとともに、共有メモリ参照情報を作成してリンク制御情報に追加する。

【0026】このようにして、作成した共有メモリ2'を、データ処理装置DPbの共有メモリアクセスタスク1bがアクセスするとき、共有メモリアクセスタスク1bは、リモートメモリ機能呼び出しによってこの共有メモリ2'をオープンする。

【0027】それにより、データ処理装置DPbのリモートメモリ管理制御部4は、記憶しているリンク制御情報の内容を参照して、指定された共有メモリ2'のノードがデータ処理装置DPaであると特定し、そのデータ処理装置DPaに対して、共有メモリ2'のリンクを依頼する。

【0028】このリンク依頼を受けたデータ処理装置DPaでは、リモートメモリ管理制御部4が、指定された共有メモリ2'のロック情報および排他モード情報を参照して、そのときのリンク依頼を受け付けるかどうかを判断し、リンク依頼を受け付ける場合には、リンクに必要な情報(アドレスやハンドル)を設定した状態で、リ

ンク受付したことをデータ処理装置D P bに返す。また、データ処理装置D P aのリモートメモリ管理制御部4は、共有メモリ2'のリンク情報をそのときのデータ処理装置D P bのノードアドレスを追加するとともに、リンクカウントの値を1つ増やす。

【0029】このときには、リンクが受け付けられたので、データ処理装置D P bのリモートメモリ管理制御部4は、通知されたリンクに必要な情報を指定した状態でリンク受付を共有メモリアクセスタスク1 bに通知する。

【0030】これにより、共有メモリアクセスタスク1 bは、その共有メモリ2'をアクセス可能な状態になる。

【0031】そして、共有メモリアクセスタスク1 bが、共有メモリ2'からデータの読み出しを行うとき、共有メモリアクセスタスク1 bは、パラメータとして共有メモリ2'のアドレス、ハンドル、および、オフセットなど、読み出すデータを指定するために必要な情報を付加した状態で、リモートメモリ管理制御機能呼び出しを実行する。

【0032】これにより、データ処理装置D P bのリモートメモリ管理制御部4は、記憶しているリンク制御情報の内容を参照して、指定された共有メモリ2'のノードがデータ処理装置D P aであると特定し、そのデータ処理装置D P aに対して、読み出しアドレスのオフセットなどを指定した状態で、共有メモリ2'からデータ読み出しを依頼する。

【0033】このデータ読み出し依頼を受けたデータ処理装置D P aでは、リモートメモリ管理制御部4が、指定された共有メモリ2'のロック情報をおよび排他モード情報を参照して、そのときのデータ読み出し依頼を受け付けるかどうかを判断し、データ読み出し依頼を受け付ける場合には、共有メモリ2'について、そのときに指定されたオフセットからデータ読み出しを行い、その読み出しデータをデータ処理装置D P bに返す。

【0034】これによって、データ処理装置D P bのリモートメモリ管理制御部4は、受信したデータを共有メモリアクセスタスク1 bに返す。

【0035】また、共有メモリアクセスタスク1 bが、共有メモリ2'にデータの書き込みを行うとき、共有メモリアクセスタスク1 bは、パラメータとして書き込みに必要な情報（共有メモリ2'でのオフセット、格納先アドレス、書き込みサイズなど）を指定し、書き込みデータを渡した状態で、リモートメモリ管理制御機能呼び出しを実行する。

【0036】これにより、データ処理装置D P bのリモートメモリ管理制御部4は、記憶しているリンク制御情報の内容を参照して、指定された共有メモリ2'のノードがデータ処理装置D P aであると特定し、そのデータ処理装置D P aに対して、オフセットなどを指定すると

ともに、書き込みデータを渡した状態で、共有メモリ2'に対するデータ書き込みを依頼する。

【0037】このデータ書き込み依頼を受けたデータ処理装置D P aでは、リモートメモリ管理制御部4が、指定された共有メモリ2'のロック情報をおよび排他モード情報を参照して、そのときのデータ書き込み依頼を受け付けるかどうかを判断し、データ書き込み依頼を受け付ける場合には、共有メモリ2'について、そのときに指定されたオフセットから、渡された書き込みデータの書き込み動作を行う。

【0038】また、共有メモリアクセスタスク1 bが、共有メモリ2'のアクセス権を放棄するとき、共有メモリアクセスタスク1 bは、パラメータとして共有メモリ2'を指定した状態で、リモートメモリ管理制御機能呼び出しを実行する。

【0039】これにより、データ処理装置D P bのリモートメモリ管理制御部4は、記憶しているリンク制御情報の内容を参照して、指定された共有メモリ2'のノードがデータ処理装置D P aであると特定し、そのデータ処理装置D P aに対して、共有メモリ2'に対するリンク終了を通知する。

【0040】このリンク終了通知を受けたデータ処理装置D P aでは、リモートメモリ管理制御部4が、共有メモリ2'のリンク情報を削除するとともに、リンクカウンタの値を1つ減らす。また、このときに、リンクカウンタの値が0になったときには、その共有メモリ2'をリンクしているプロセスがない場合なので、共有メモリ2'を廃棄する。

【0041】また、共有メモリアクセスタスク1 bが、共有メモリ2'のロックを実行したいとき、共有メモリアクセスタスク1 bは、パラメータとしてロックする範囲（アドレス範囲指定など）や排他モード情報を指定した状態で、共有メモリ2'について、リモートメモリ管理制御機能呼び出しを実行する。

【0042】これにより、データ処理装置D P bのリモートメモリ管理制御部4は、記憶しているリンク制御情報の内容を参照して、指定された共有メモリ2'のノードがデータ処理装置D P aであると特定し、そのデータ処理装置D P aに対して、ロックする範囲や排他モード情報を指定した状態で、共有メモリ2'に対するロック依頼を通知する。

【0043】このロック依頼を受けたデータ処理装置D P aでは、リモートメモリ管理制御部4が、指定された共有メモリ2'のロック情報をおよび排他モード情報を参照して、そのときのロック依頼を受け付けるかどうかを判断し、ロック依頼を受け付ける場合には、共有メモリ2'について、そのときに指定された範囲について、ロックまたは排他モード情報の設定を行う。

【0044】このようにして、データ処理装置D P aに共有メモリ2'が作成され、この共有メモリ2'を他の

データ処理装置D P b からアクセスすることができ、また、共有メモリ2' が廃棄される。

【0045】また、データ処理装置D P b の共有メモリアクセスタスク1 b は、ローカルメモリをアクセスするときと同様のリモートメモリ管理制御機能呼び出しを実行することで、他のデータ処理装置D P a の共有メモリ2' をアクセスすることができるので、共有メモリアクセスタスク1 b は、ローカルメモリと共有メモリ（リモートメモリ）の区別をすることなく、共有メモリ2' をアクセスすることができる。

【0046】図7は、共有メモリを作成するノードが実行する処理の一例を示している。

【0047】まず、リモートメモリ管理制御部4 に対して、共有メモリの作成を依頼する（処理101）。これにより、リモートメモリ管理制御部4 は、共有メモリを確保するとともに、それをアクセスするために必要な管理情報を作成する（処理102）。

【0048】そして、いずれかのタスクが共有メモリに対するデータ書き込みを要求すると（処理103）、リモートメモリ管理制御部4 は、指定された共有メモリに対して、データの書き込むを実行する（処理104）。

【0049】図8は、他のノードの共有メモリから、データを読み出すときの処理の一例を示している。

【0050】まず、リモートメモリ管理制御部4 に対して、共有メモリのオープンを依頼する（処理201）。これにより、リモートメモリ管理制御部4 は、指定された共有メモリを保持するノードを特定するとともに、その共有メモリをアクセスするための管理情報を作成する（処理202）。

【0051】次に、共有メモリからのデータの読み出しが要求すると（処理203）、リモートメモリ管理制御部4 は、目的ノードの共有メモリからデータを読み出して、その読み出しデータを受信する（処理204）。

【0052】図9は、共有メモリの作成時の他の処理例を示している。

【0053】まず、共有メモリを作成するノードでは、管理情報を作成し（処理301）、共有メモリ作成を他のノードに通知する（処理302）。次いで、共有メモリの作成タスクに対して、参照情報を返し（処理303）、共有メモリ一覧情報の内容を更新する（処理304）。

【0054】また、共有メモリを作成しないノードでは、図10に示したように、共有メモリ作成通知を受信すると（処理401）、そのときの共有メモリ作成ノードのノードアドレスを共有メモリ一覧情報に追加する（処理402）。

【0055】図11は、共有メモリのオープン時に、共有メモリとのリンクを要求するノードが実行する処理例を示している。

【0056】まず、そのときに共有メモリ一覧情報を保

持しているかどうかを調べる（判断501）。判断501の結果がNOになるときには、各ノードに対して、共有メモリを保持しているかどうかを問い合わせ（処理502）、各ノードからの応答を受信し（処理503）、その受信結果に基づいて、共有メモリ一覧情報を作成する（処理504）。

【0057】処理504を終了したとき、および、判断501の結果がYESになるときには、判断505に移行する。この判断505では、共有メモリ一覧情報に1つ以上のノードが登録されているかどうかを調べ、判断505の結果がYESになるときは、そのノードに対して、リンク依頼を発行して（処理506）、そのリンク依頼の結果を受信する（処理507）。

【0058】そして、そのときにリンクが成功したかどうかを調べて（処理506）、判断506の結果がYESになるときには、共有メモリ参照情報を作成して、保存する（処理507）。

【0059】図11は、共有メモリ有無の問い合わせを受信したときの処理例を示している。

【0060】共有メモリ有無の問い合わせを受信すると（処理601）、そのときの共有メモリの有無状態を、その問い合わせ元のノードに通知する（処理602）。

【0061】図12は、リンク依頼を受信したときの処理例を示す。

【0062】リンク依頼を受信すると（処理701）、その依頼元について、アクセス権等をチェックして（処理702）、リンク可能であるかどうかを調べる（判断703）。

【0063】判断703の結果がYESになるときは、リンク情報を応答して（処理704）、リンクカウンタの値を1つ増やす（処理705）。判断703の結果がNOになるときには、リンク不可を通知する（処理706）。

【0064】図14は、共有メモリをアクセスするときの、要求側のノードの処理の一例を示している。

【0065】まず、タスクから指定されたパラメータを参照して、共有メモリ参照情報を得る（処理801）。このときに指定されたパラメータが正常値であり、正しく共有メモリ参照情報を得た場合には（判断802の結果がYES）、共有メモリを保持するノードのリモートメモリ管理制御サーバ機能に対して、アクセス要求を発行する（処理803）。

【0066】次に、そのリモートメモリ管理制御サーバ機能からの応答を受信し（処理804）、そのときにアクセス可が通知されたかどうかを調べる（判断805）。判断805の結果がYESになるときには、そのときにデータ読み出しを実行するかどうかを調べる（判断806）。

【0067】判断806の結果がYESになるときには、リモートメモリ管理制御サーバ機能にデータ読み出

し要求を発行し（処理808）、それによって受信したデータをアクセス要求したタスクに渡す（処理809）。

【0068】データ書き込みが要求された場合で、判断806の結果がNOになるときは、リモートメモリ管理制御サーバ機能にデータ書き込み要求を発行し（処理810）、タスクから渡されたデータを送出する（処理811）。

【0069】図15は、アクセス要求されたときの処理の一例を示している。

【0070】アクセス要求を受信すると（処理901）、アクセス権等をチェックし（処理902）、アクセス可能であるかどうかを調べる（判断903）。判断903の結果がNOになるときには、アクセス要求元に対してアクセス不可を通知する（処理904）。

【0071】判断903の結果がYESになるときには、アクセス要求元に対してアクセス可を通知し（処理905）、そのときにデータ読み出しが通知されたかどうかを調べる（判断906）。

【0072】判断906の結果がYESになるときには、指定されたデータ読み出し動作を実行して（処理907）、その読み出したデータをアクセス要求元に転送する（処理908）。

【0073】また、判断906の結果がNOになるときには、書き込みデータを受信して（処理909）、その受信データを、指定した態様で書き込む（処理910）。

【0074】図16は、リンク解放が指令されたときの処理例を示す。

【0075】タスクから指定された共有メモリを保持するノードのリモートメモリ管理制御サーバ機能にリンク解放要求を発行し（処理1001）、共有メモリ参照情報から、その指定された共有メモリの情報を削除する（処理1002）。

【0076】図17は、リンク解放要求を受信したときの処理の一例を示している。

【0077】リンク解放要求を受信すると（処理1101）、その要求元のノードをリンク情報から消去して（処理1102）、リンクカウンタの値を1つ減らす（処理1103）。ここで、リンクカウンタの値が0になっているかどうかを調べ（判断1104）、判断1104の結果がYESになるときには、共有メモリおよび管理情報を廃棄する（処理1105）。

【0078】図18は、リンク解放が指令されたときの他の例を示している。

【0079】タスクから共有メモリの廃棄が指令されると（処理1201）、そのときに共有メモリにリンクしているノードに対して共有メモリの廃棄通知を発行し（処理1202）、各ノードからの応答を受信する（処理1203）。

【0080】全てのノードが廃棄可能を応答した場合で、共有メモリを廃棄できるかどうかを調べ（判断1204）、判断1204の結果がYESになるときには、共有メモリおよび管理情報を廃棄する（処理1205）。

【0081】図19は、共有メモリの廃棄通知を受信したときに実行する処理の一例を示している。

【0082】共有メモリ廃棄通知を受信すると（処理1301）、そのときに通知された共有メモリを廃棄可能な状態になっているかどうかを調べ（判断1302）、判断1302の結果がYESになるときは、廃棄可を通知するとともに、共有メモリ参照情報を廃棄する（処理1303）。また、判断1302の結果がNOになるときには、廃棄不可を通知する（処理1304）。

【0083】ところで、各データ処理装置DPa～DPkで保持する共有メモリSM1～SMkを、仮想的な線形メモリの一部に配置することができる。

【0084】その場合、共有メモリ管理情報は、図20(a)に示すように、データオブジェクトを保存したアドレス、データオブジェクトのサイズ、仮想的な線形メモリ領域でこの共有メモリが占める領域の先頭アドレスをあらわすオフセット、このデータオブジェクトをリンクしているノードの一覧からなるリンク情報、このデータオブジェクトをリンクしているタスクの数をあらわすリンクカウンタ、このデータオブジェクトがロックされているかどうかをあらわすロック情報、および、このデータオブジェクトに対するアクセス制限などをあらわす排他モード情報から構成する。

【0085】また、同図(b)に示したように、各データ処理装置DPa～DPkには、仮想的線形メモリにおける共有メモリの配置状況を記憶するための仮想線形メモリマップを形成する。この仮想線形メモリマップは、おのおのの共有メモリに関する複数の共有メモリマップからなる。また、おのおのの共有メモリマップの内容は、同図(c)に示したように、オフセットと、ノードアドレスと、サイズからなる。

【0086】したがって、この場合には、複数ノード上の複数の共有メモリを、連続したメモリとしてアクセスすることができ、共有メモリをアクセスするときの処理の負担を軽減できる。

【0087】この場合、タスクがリモートメモリ管理制御機能呼び出しを実行して、共有メモリの作成を要求したときの処理例を図21に示す。

【0088】共有メモリの作成が要求されると、まず、そのときに仮想線形メモリマップを保持しているかどうかを調べる（判断1401）。判断1401の結果がNOになるときには、各ノードに対して、保持している共有メモリに関し、共有メモリに関するマップ情報（オフセット、サイズなど）の通知を要求し（処理1402）、おのおののノードからの通知データを受信し（処

11

理1403)、そのときの受信データに基づいて、仮想線形メモリマップを作成する(処理1404)。

【0089】処理1404を終了したとき、および、判断1401の結果がYESになるときには、共有メモリを実メモリに確保できるかどうかを調べる(判断1405)。

【0090】判断1405の結果がYESになるときには、仮想線形メモリマップに基づき、仮想線形メモリにおいて、作成する共有メモリのオフセット値を決定し、その共有メモリの管理情報を作成する(処理1406)。このとき、最も高位のアドレスに位置するブロックの次の位置をオフセットとしたり、空き領域に共有メモリを確保することができる。

【0091】そして、その作成した共有メモリについて、共有メモリを作成した旨、および、共有メモリに関する情報を各ノードに通知する(処理1407)。

【0092】図22は、共有メモリのマップ情報通知要求を受信したときに実行する処理の一例を示している。

【0093】共有メモリのマップ情報通知要求を受信すると(処理1501)、そのときに自ノードで保持している共有メモリのマップ情報を、その要求ノードに対して送信する(処理1502)。

【0094】図23は、共有メモリ作成通知を受信したときに実行する処理の一例を示している。

【0095】共有メモリ作成通知を受信すると(処理1601)、そのときに通知された内容で、仮想線形メモリマップの内容を更新する(処理1602)。

【0096】タスクがリモートメモリ管理制御機能呼び出しを実行して、共有メモリへのアクセスを要求したときの処理例を図24に示す。このとき、アクセス要求する共有メモリのオフセットなど、共有メモリを指定する情報がパラメータとして指定される。

【0097】共有メモリへのアクセス要求を受信すると、まず、そのときに仮想線形メモリマップを保持しているかどうかを調べる(判断1701)。判断1701の結果がNOになるときには、各ノードに対して、保持している共有メモリに関し、共有メモリに関するマップ情報(オフセット、サイズなど)の通知を要求し(処理1702)、おのののノードからの通知データを受信し(処理1703)、そのときの受信データに基づいて、仮想線形メモリマップを作成する(処理1704)。

【0098】処理1704を終了したとき、および、判断1701の結果がYESになるときには、そのときに指定されたオフセットに共有メモリが存在するかどうかを調べる(判断1705)。

【0099】判断1705の結果がYESになるときには、共有メモリを保持するノードに対してリンクを要求し(処理1706)、そのノードからの応答を受信する(処理1707)。

10

12

【0100】このときに、リンクが成功したかどうかを調べて(判断1708)、判断1708の結果がYESになるときには、共有メモリ参照情報を作成する(処理1709)。

【0101】図25は、リンク要求を受信したときの処理例を示している。

【0102】まず、そのときのリンク要求元について、アクセス権をチェックし(処理1801)、リンク可能であるかどうかを調べる(判断1802)。判断1802の結果がYESになるときには、共有メモリに関するリンク情報を作成して、リンク要求元に送出する(処理1803)。また、判断1802の結果がNOになるときには、リンク不可をリンク要求元に通知する(処理1804)。

【0103】ところで、共有メモリに一意の識別子を付加すると、この識別子を指定するだけでネットワーク上の複数の共有メモリのうち、目的のものを指定することができるので、共有メモリの指定が簡単になる。

【0104】その場合、共有メモリ管理情報は、図26(a)に示すように、データオブジェクトを保存したアドレス、データオブジェクトのサイズ、この共有メモリに付加された識別子、このデータオブジェクトをリンクしているノードの一覧からなるリンク情報、このデータオブジェクトをリンクしているタスクの数をあらわすリンクカウンタ、このデータオブジェクトがロックされているかどうかをあらわすロック情報、および、このデータオブジェクトに対するアクセス制限などをあらわす排他モード情報から構成する。

【0105】また、同図(b)に示したように、各データ処理装置DPa～DPkには、共有メモリと識別子との関係をあらわす共有メモリ識別子テーブルを形成する。この共有メモリ識別子テーブルは、おののの共有メモリに関する共有メモリ識別子情報からなる。また、おののの共有メモリ識別子情報の内容は、同図(c)に示したように、識別子と、ノードアドレスからなる。

【0106】この場合、タスクがリモートメモリ管理制御機能呼び出しを実行して、付加する識別子を指定した状態で共有メモリの作成を要求したときの処理例を図27に示す。

【0107】共有メモリの作成が要求されると、まず、そのときに共有メモリ識別子テーブルを保持しているかどうかを調べる(判断1901)。判断1901の結果がNOになるときには、各ノードに対して、保持している共有メモリに関し、共有メモリの識別子を問い合わせ(処理1902)、おのののノードからの通知データを受信し(処理1903)、そのときの受信データに基づいて、共有メモリ識別子テーブルを作成する。

【0108】処理1903を終了したあと、および、判断1901の結果がYESになるときには、そのときに指定された識別子が、既に他の共有メモリに付加されて

いるかどうかを調べる（判断1904）。

【0109】判断1904の結果がYESになるときには、そのときに新たに共有メモリを作成できるかどうかを調べ（判断1905）、判断1905の結果がYESになるときには、その作成した共有メモリについて、共有メモリを作成した旨、および、共有メモリに関する情報を各ノードに通知する（処理1906）。

【0110】図28は、共有メモリの識別子問い合わせを受信したときに実行する処理の一例を示している。

【0111】共有メモリの識別子問い合わせを受信すると（処理2001）、そのときに自ノードで保持している共有メモリの識別子を、その要求ノードに対して送信する（処理2002）。

【0112】図29は、共有メモリ作成通知を受信したときに実行する処理の一例を示している。

【0113】共有メモリ作成通知を受信すると（処理2101）、そのときに通知された内容で、共有メモリ識別子テーブルの内容を更新する（処理2102）。

【0114】タスクがリモートメモリ管理制御機能呼び出しを実行して、共有メモリへのアクセスを要求したときの処理例を図30に示す。このとき、アクセス要求する共有メモリの識別子など、共有メモリを指定する情報がパラメータとして指定される。

【0115】共有メモリへのアクセス要求を受信すると、まず、そのときに共有メモリ識別子テーブルを保持しているかどうかを調べる（判断2201）。判断2201の結果がNOになるときには、各ノードに対して、保持している共有メモリに関し、共有メモリの識別子を問い合わせ（処理2202）、おのおののノードからの通知データを受信し（処理2203）、そのときの受信データに基づいて、共有メモリ識別子テーブルを作成する。

【0116】処理2203を終了したあと、および、判断2201の結果がYESになるときには、そのときに指定された識別子が、共有メモリ識別子テーブルに含まれているかどうかを調べる（判断2204）。

【0117】判断2204の結果がYESになるときには、共有メモリを保持するノードに対してリンクを要求し（処理2205）、そのノードからの応答を受信する（処理2206）。

【0118】このときに、リンクが成功したかどうかを調べて（判断2207）、判断2207の結果がYESになるときには、共有メモリ参照情報を作成する（処理2208）。

【0119】さて、本発明は、仮想記憶管理機構を備えたノードを用いる場合についても同様にして適用することができる。

【0120】図31は、仮想記憶管理機構を備えたデータ処理装置のシステム構成の一例を示している。なお、同図では、本発明の要部に関する部分のみを示してお

り、他の部分については、説明を省略する。また、同図において、図2と同一部分および相当する部分には、同一符号を付している。

【0121】同図において、タスク群1'は、このデータ処理装置DPで実行されるユーザプロセスを構成するものであり、メモリ2は、タスク群1の実行時に用いられるメモリ領域であり、メモリ管理制御部3'は、仮想メモリ管理制御部（仮想記憶管理機構）6またはリモートメモリ管理制御部4'が、メモリ2に対してデータアクセスする動作を管理するためのものである。

【0122】リモートメモリ管理制御部4'は、メモリ2に形成する共有メモリSM（SM1～SMk）のデータアクセスのためのものであり、自ノードの共有メモリSMに他ノードからアクセスするためのサーバ（RMMS／サーバ）機能と、他ノードの共有メモリSMにアクセスするためのクライアント（RMMS／クライアント）機能を備えている。

【0123】仮想メモリ管理制御部6は、仮想記憶管理機構を実現するためのものであり、タスク群1'は、この仮想メモリ管理制御部6を介して、メモリ2にアクセスする。

【0124】この場合、メモリ2のデータアクセスするために、図32(a)に示したようなディスクリプタが用いられる。このディスクリプタは、おのおののメモリオブジェクトをアクセスするためのメモリ情報からなる。

【0125】また、このメモリ情報は、同図(b)に示したように、対応する記憶領域の物理アドレスを示すアドレス、サイズ、リモートメモリであるかローカルメモリであるかを区別するためのリモート／ローカル情報、データが2次記憶装置にスワップアウトされているか否かをあらわすスワップ情報、このデータオブジェクトをリンクしているノードの一覧からなるリンク情報、このデータオブジェクトをリンクしているタスクの数をあらわすリンクカウンタ、このデータオブジェクトがロックされているかどうかをあらわすロック情報、および、このデータオブジェクトに対するアクセス制限などをあらわす排他モード情報からなる。

【0126】この場合、タスクがリモートメモリ管理制御機能呼び出しを実行して、共有メモリに対するリンクを要求するときの処理例を図33に示す。

【0127】タスクが、リモートメモリ管理制御機能呼び出しを実行して、共有メモリに対するリンクを要求すると（処理2301）、リモートメモリ管理制御機構からは、共有メモリアクセスのための仮想アドレス情報（メモリ情報）が返るので（処理2302）、タスクは、この仮想アドレス情報を保存する。

【0128】図34は、リモートメモリ管理制御機能呼び出しで、共有メモリに対するリンクが要求されたときの、リモートメモリ管理制御部4'およびシステム側の

処理例を示す。

【0129】まず、リモートメモリ管理制御部4'は、リンク情報を作成し(処理2401)、リモートモードの対応するディスクリプタ(メモリ情報)を作成する(処理2402)。そして、そのディスクリプタに対応した仮想アドレス情報をタスクに返す(処理2403)。

【0130】また、タスクがリモートメモリ管理制御機能呼び出しを実行して、共有メモリに対するアクセスを要求するときの処理例を図35に示す。

【0131】まず、タスクは、そのときにシステム(リモートメモリ管理制御機構)から返された仮想アドレス情報を用いて、リモートメモリ管理制御機構に対して、共有メモリのアクセスを要求する(処理2501)。

【0132】これにより、システムからアクセス可が通知されると(判断2502の結果がYES)、そのときにアクセス要求した共有メモリに対して、適宜なデータ処理を実行する(処理2503)。

【0133】図36は、タスクからメモリアクセス要求されたときに、システムが実行する処理の一例を示している。

【0134】まず、タスクから指定された仮想アドレスから、対応するディスクリプタ(メモリ情報)を得る(処理2601)。そして、そのアクセス対象がリモートメモリであるかどうかを調べる(判断2602)。

【0135】判断2602の結果がNOになるときは、そのタスクに関するアクセス権をチェックして(処理2603)、アクセス可であるかどうかを調べる(判断2604)。判断2604の結果がNOになるときは、アクセス要求したタスクに対して、アクセス不可を通知する(処理2605)。

【0136】また、判断2604の結果がYESになるときは、そのときのメモリオブジェクトがスワップアウトされているかどうかを調べ(判断2606)、判断2606の結果がYESになるときには、そのデータを実メモリ上にロードする(処理2607)。

【0137】このようにして、データアクセス可能な状態になると、タスクに対して、データアクセス処理を実行させる(処理2608)。

【0138】また、アクセス要求された対象がリモートメモリ(共有メモリ)であり、判断2602の結果がYESになるときには、そのときに得たディスクリプタに基づいて、共有メモリをアクセスするためのリンク情報を形成し(処理2609)、その共有メモリを保持するノードのリモートメモリ管理制御サーバに対して、アクセス要求を発行する(処理2610)。

【0139】そのときのアクセス要求に対して、アクセス可が通知されたときには(判断2611の結果がYES)、処理2608に移行し、タスクに対してデータアクセス処理を実行させる。また、アクセス不可が通知さ

れたときで、判断2611の結果がNOになるときは、処理2605に移行し、タスクに対してアクセス不可を通知する。

【0140】したがって、本実施例では、仮想記憶管理機構を利用することで、タスクは、共有メモリに対するアクセスを、ローカルメモリに対するアクセスと区別することなく実行できるので、タスク側の処理の負担がより軽減する。

【0141】なお、本発明は、上述したようなシステム以外のものについても、同様にして適用することができる。

#### 【0142】

【発明の効果】以上説明したように、本発明によれば、異なるノード間でメモリを共有できるので、ネットワークシステム全体でのデータの共有を、容易に実現することができる。また、仮想記憶管理機構を備えている場合には、共有メモリ空間を仮想記憶管理機構を用いて直接アクセスすることができるので、データのアクセス効率がよく、また、共有メモリを高速にアクセスすることができるという効果を得る。

#### 【図面の簡単な説明】

【図1】本発明の一実施例にかかるローカルエリアネットワークシステムを示したブロック図。

【図2】データ処理装置(ノード)の一例を示したブロック図。

【図3】メモリ領域の分割の一例を示した概略図。

【図4】共有メモリ領域に記憶されるデータの一例を示した概略図。

【図5】ローカルメモリ領域に記憶されるデータの一例を示した概略図。

【図6】共有メモリに対するアクセスなどの種々の処理動作を説明するためのモデルブロック図。

【図7】共有メモリを形成するノードの処理例を示したフローチャート。

【図8】共有メモリを形成しない他のノードの処理例を示したフローチャート。

【図9】共有メモリ形成時の処理例を示したフローチャート。

【図10】共有メモリ作成通知を受信したときの処理例を示したフローチャート。

【図11】共有メモリにリンク依頼するときの処理例を示したフローチャート。

【図12】共有メモリ有無問い合わせを受信したときの処理の一例を示したフローチャート。

【図13】リンク依頼を受信したときの処理の一例を示したフローチャート。

【図14】共有メモリにアクセス要求するときの処理例を示したフローチャート。

【図15】アクセス要求を受信したときの処理例を示したフローチャート。

【図16】共有メモリのリンク解放を要求したときの処理例を示したフローチャート。

【図17】リンク解放通知を受信したときの処理例を示したフローチャート。

【図18】共有メモリの廃棄要求時の処理例を示したフローチャート。

【図19】共有メモリ廃棄通知を受信したときの処理例を示したフローチャート。

【図20】本発明の他の実施例にかかる管理情報などの一例を示した概略図。

【図21】共有メモリの作成要求時の他の処理例を示したフローチャート。

【図22】共有メモリマップ通知要求を受信したときの処理の一例を示したフローチャート。

【図23】共有メモリ情報を受信したときの処理例を示したフローチャート。

【図24】共有メモリにリンク依頼するときの処理の他の例を示したフローチャート。

【図25】リンク依頼を受信したときの処理例を示したフローチャート。

【図26】本発明のさらに他の実施例にかかる管理情報などの一例を示した概略図。

【図27】共有メモリの作成要求時のさらに他の処理例\*

【図28】共有メモリ識別子の問い合わせを受信したときの処理例を示したフローチャート。

【図29】共有メモリ情報を受信したときの処理例を示したフローチャート。

【図30】リンク依頼時のさらに他の処理例を示したフローチャート。

【図31】本発明の別の実施例にかかるデータ処理装置を示したブロック図。

【図32】図31の装置のディスクリプタの一例を示した概略図。

【図33】共有メモリオープン要求時のタスクの処理例を示したフローチャート。

【図34】共有メモリオープン要求時のシステムの処理例を示したフローチャート。

【図35】共有メモリアクセス時のタスクの処理例を示したフローチャート。

【図36】共有メモリアクセス時のシステムの処理例を示したフローチャート。

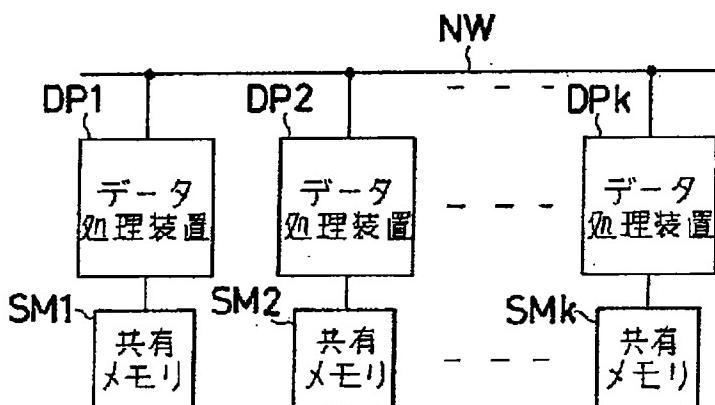
#### 【符号の説明】

NW ネットワーク

DP1～DPk, DP データ処理装置

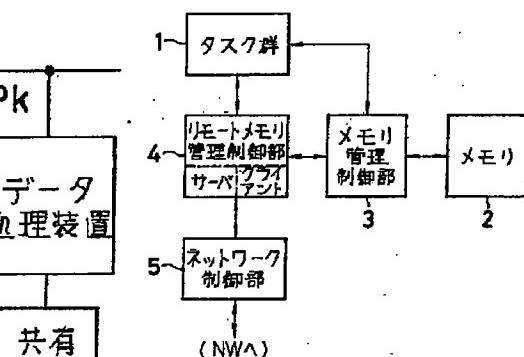
SM1～SMk, SM 共有メモリ

【図1】



【図3】

【図2】

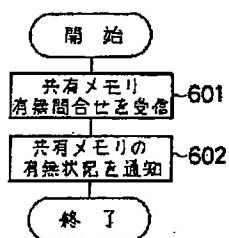
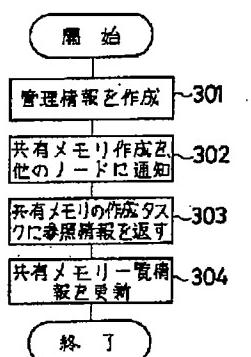
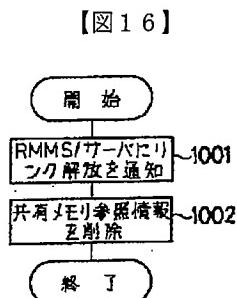
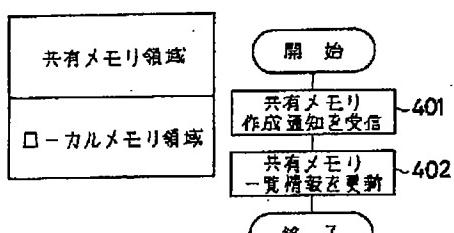


【図10】

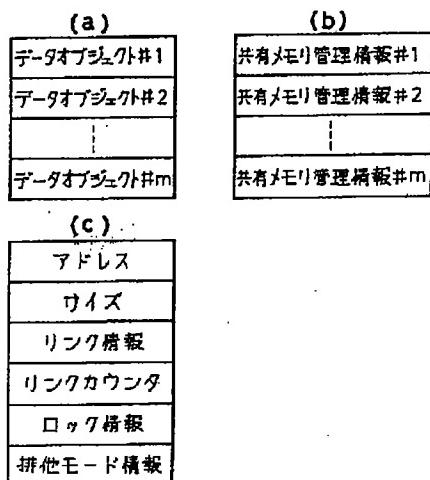
【図16】

【図9】

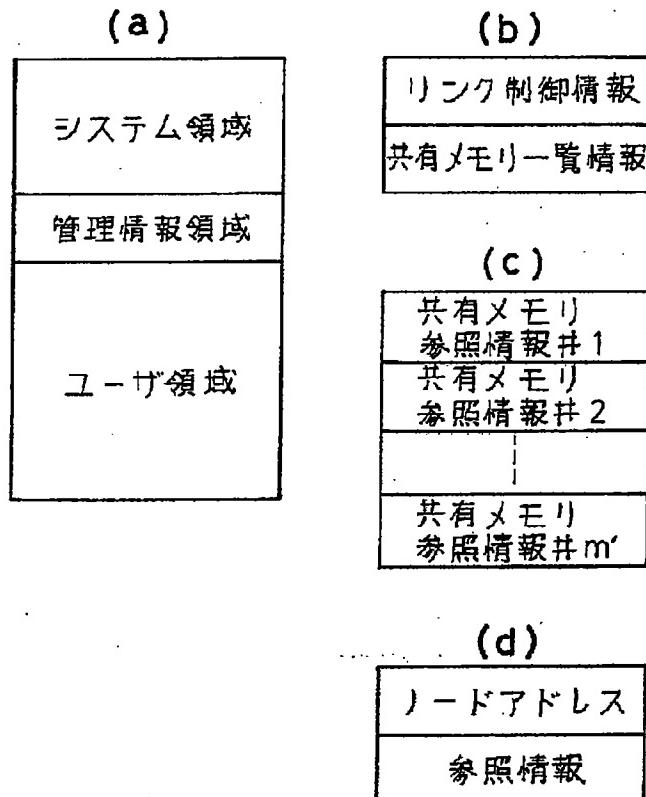
【図12】



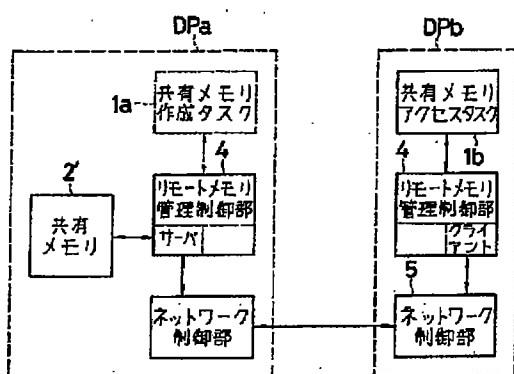
【図4】



【図5】



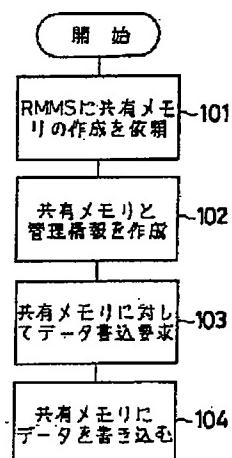
【図6】



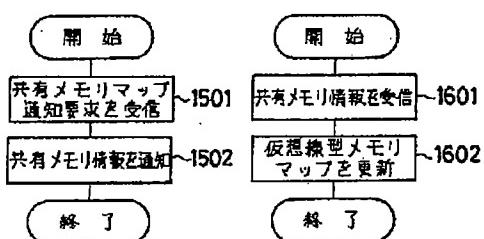
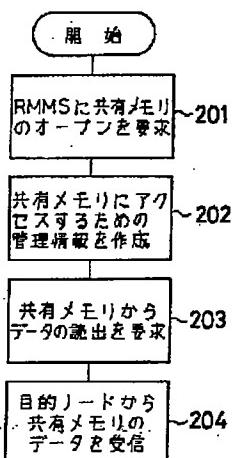
【図22】

【図23】

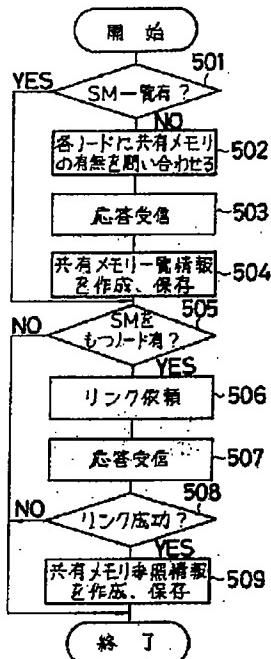
【図7】



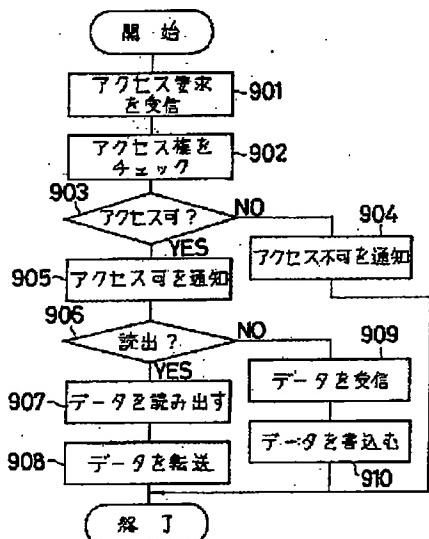
【図8】



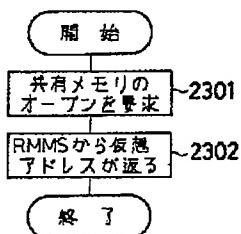
【図11】



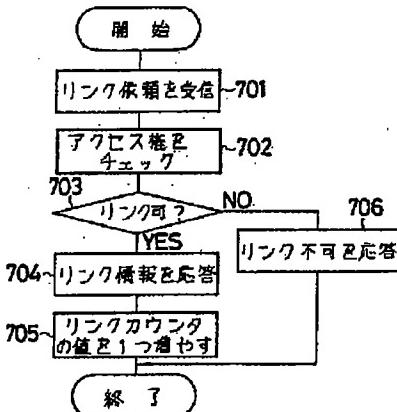
【図15】



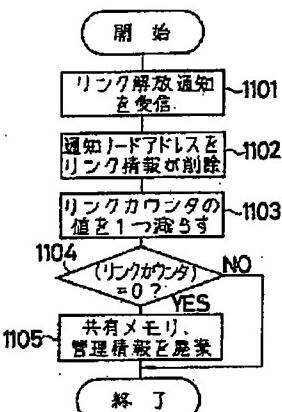
【図33】



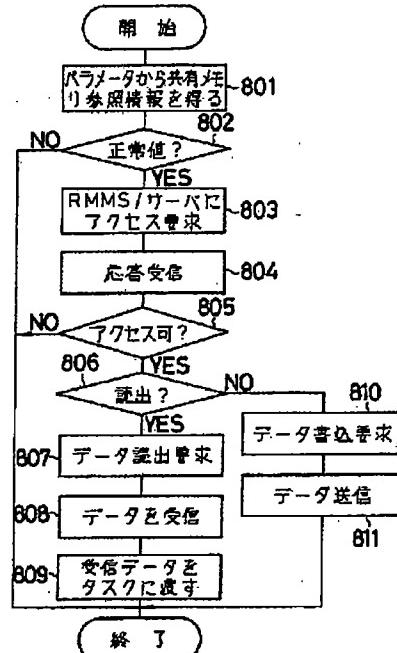
【図13】



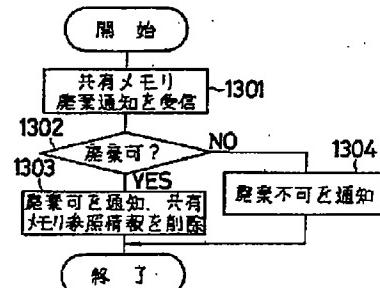
【図17】



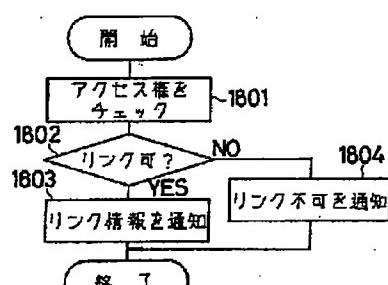
【図14】



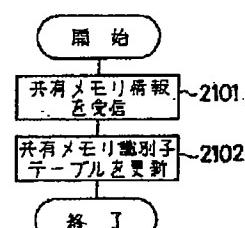
【図19】



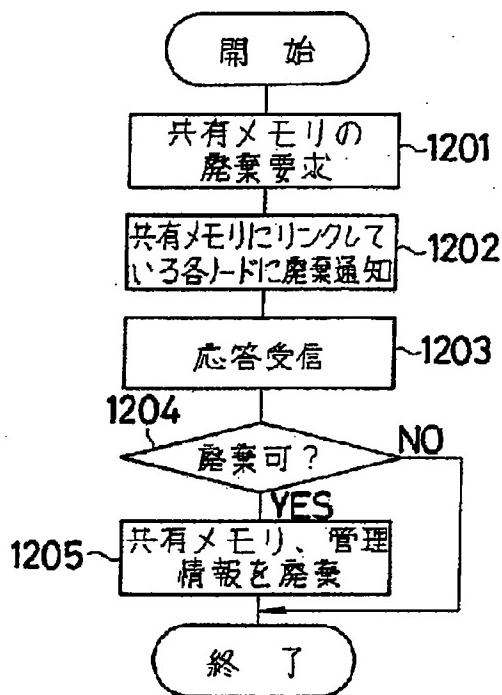
【図25】



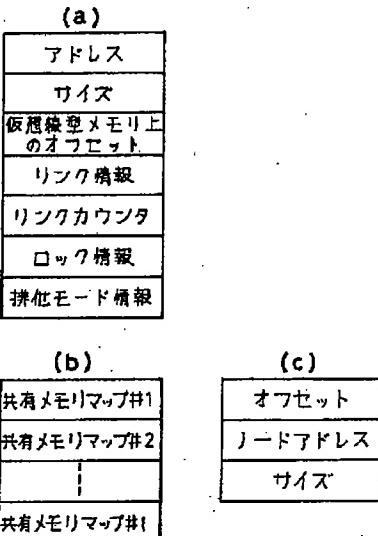
【図29】



【図18】



【図20】



【図26】

(a)

アドレス
サイズ
識別子
リンク情報
リンクカウント
ロック情報
併合モード情報

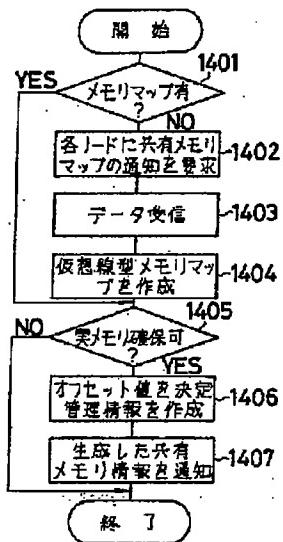
(b)

共有メモリ 識別子情報#1
共有メモリ 識別子情報#2
⋮
共有メモリ 識別子情報#n

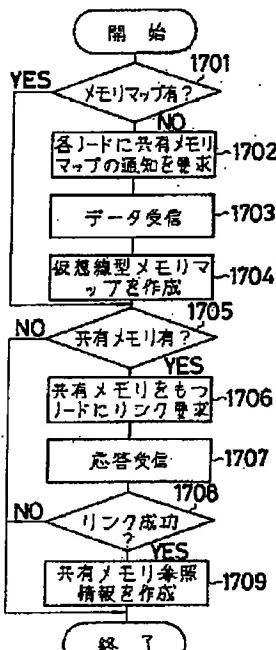
(c)

識別子
ノードアドレス

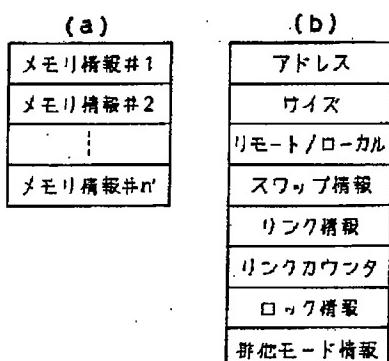
【図21】



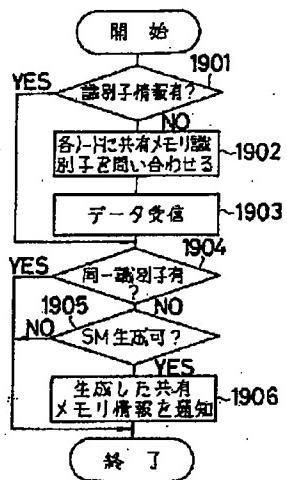
【図24】



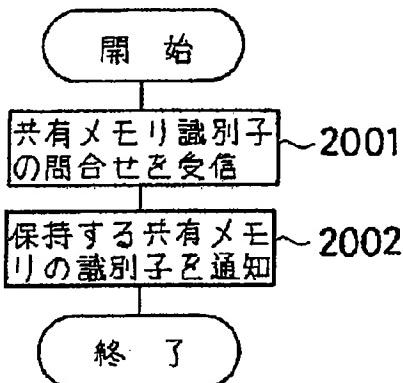
【図32】



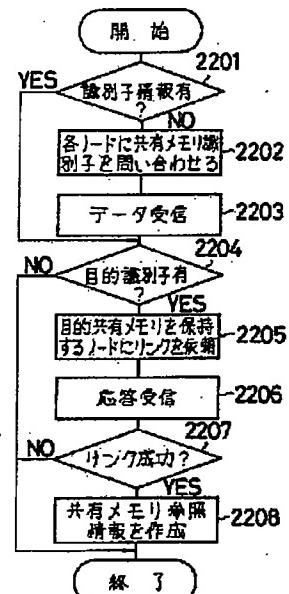
【図27】



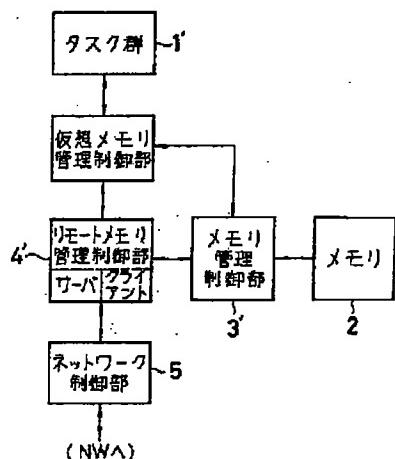
【図28】



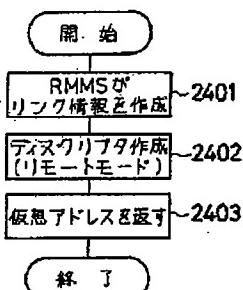
【図30】



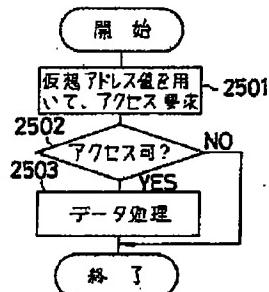
【図31】



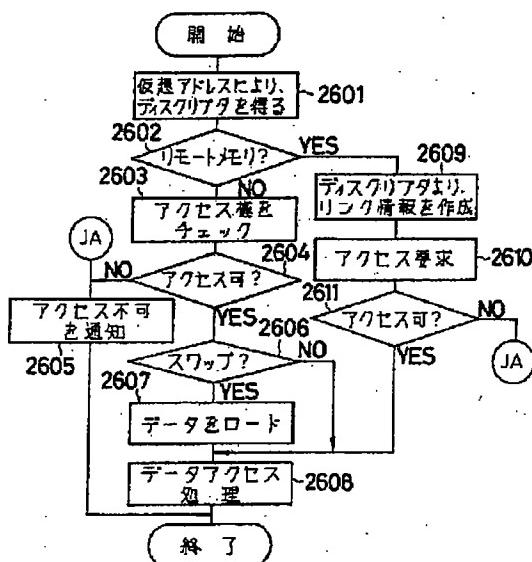
【図34】



【図35】



【図36】



# PATENT ABSTRACTS OF JAPAN

(11)Publication number : 07-084921

(43)Date of publication of application : 31.03.1995

(51)Int.Cl. G06F 13/00  
H04L 12/40

(21)Application number : 05-183145

(71)Applicant : RICOH CO LTD

(22)Date of filing : 30.06.1993

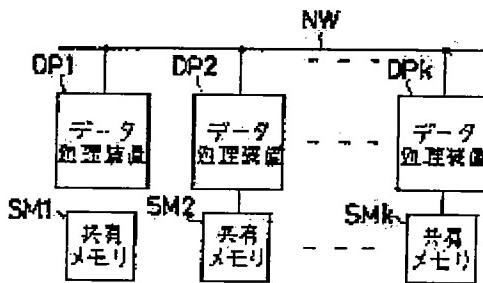
(72)Inventor : MORITA TAKEHIKO

## (54) NETWORK SYSTEM

### (57)Abstract:

**PURPOSE:** To obtain a system which smoothly and flexibly enables data and memories to be shared by providing each node with a common memory area wherein a memory object which can be used in common by other nodes can be saved.

**CONSTITUTION:** Plural data processors (node) DP1-DPk such as personal computers and work stations are connected through a network NW. The respective data processors DP1-DPk are provided with common memories SM1-SMk. The network NW is equipped with data communication functions such as a one-to-one two-way communication function and a broadcast function which does not specify an address. Node addresses to be specified on the network NW are set in the respective data processors DP1-DPk. Further, the transmission and reception of data to and from the respective data processors DP1-DPk are actualized by a server/client model. Consequently, the different nodes can share the memories.



## LEGAL STATUS

[Date of request for examination]

[Date of sending the examiner's decision of rejection]

[Kind of final disposal of application other than the examiner's decision of rejection or application converted registration]

[Date of final disposal for application]

[Patent number]

[Date of registration]

[Number of appeal against examiner's decision of rejection]

[Date of requesting appeal against examiner's decision of rejection]

[Date of extinction of right]

**\* NOTICES \***

JPO and INPIT are not responsible for any damages caused by the use of this translation.

- 1.This document has been translated by computer. So the translation may not reflect the original precisely.
- 2.\*\*\*\* shows the word which can not be translated.
- 3.In the drawings, any words are not translated.

---

**CLAIMS**

---

**[Claim(s)]**

[Claim 1] The network system characterized by preparing the shared memory field where the memory object which can be shared from other nodes to each node is saved in the network system which connected two or more nodes.

[Claim 2] Said shared memory field is a network system according to claim 1 characterized by constituting a part of imagination linearity memory area defined in the network system.

[Claim 3] It is the network system according to claim 1 characterized by adding a unique identifier to said memory object saved to said shared memory field, and being able to access from each node using the identifier.

[Claim 4] The network system characterized by attaching the virtual-memory-management address corresponding to the above-mentioned virtual-memory memory control unit in the shared memory field while each prepares the shared memory field where the memory object which can be shared from other nodes to each node in the network system which connected two or more nodes equipped with the virtual-memory-management device is saved.

---

[Translation done.]

**\* NOTICES \***

JPO and INPI are not responsible for any damages caused by the use of this translation.

1. This document has been translated by computer. So the translation may not reflect the original precisely.
- 2.\*\*\*\* shows the word which can not be translated.
3. In the drawings, any words are not translated.

---

**DETAILED DESCRIPTION**

---

**[Detailed Description of the Invention]****[0001]**

[Industrial Application] This invention relates to the network system which connects two or more nodes.

**[0002]**

[Description of the Prior Art] For example, through networks, such as a Local Area Network, two or more nodes (data processor), such as a personal computer and a workstation, are connected, and the following is conventionally used as an approach of sharing data objects (data in which a file and activation are possible) among two or more of these nodes.

[0003] For example, there is an approach of exchanging a required data object by performing the communication link of 1 to 1 between two nodes, a correspondence procedure which prepares the posts for data reception (mail box etc.), and transmits a data object from a data source to this post. By the approach of this latter, the communication link of one-pair \*\* is realizable. Moreover, there is also an approach using the broadcasting function of a Local Area Network as a correspondence procedure of one-pair \*\*.

**[0004]**

[Problem(s) to be Solved by the Invention] However, in such the former, in order that the share approach of data may mainly take the format of transmission and reception of data, un-arranging [that need to be processed / of a phase hand node / for specification and data are transmitted also to phase hand nodes other than the purpose] arises.

[0005] Moreover, when the whole network system shares data and it manages a system smoothly, the conventional approach mentioned above is not suitable for application instead of the existing communication link of wanting to only share memory and data between the whole network system of directivity which specifies a phase hand node.

[0006] This invention is made in view of this actual condition, and aims at offering the network system which can realize sharing of data or memory smoothly and flexibly.

**[0007]**

[Means for Solving the Problem] This invention prepares the shared memory field where the memory object which can be shared from other nodes to each node is saved in the network system which connects two or more nodes. Moreover, said shared memory field constitutes a part of imagination linearity memory area defined within the network system. Moreover, it is good for said memory object saved to said shared memory field to add a unique identifier and to enable it to access from each node using the identifier.

[0008] Moreover, while each prepares the shared memory field where the memory object which can be shared from other nodes to each node is saved in the network system which connects two or more nodes equipped with the virtual-memory-management device, the virtual-memory-management address corresponding to the above-mentioned virtual-memory memory control unit is added to the shared memory field.

**[0009]**

[Function] Therefore, since memory is sharable between different nodes, sharing of the data in the whole network system is easily realizable. Moreover, since direct access of the shared

memory space can be carried out using a virtual-memory-management device when it has the virtual-memory-management device, the access efficiency of data is good and can access a shared memory at a high speed.

[0010]

[Example] Hereafter, the example of this invention is explained to a detail, referring to an accompanying drawing.

[0011] Drawing 1 shows the network system concerning one example of this invention.

[0012] In this drawing, two or more data processors (node) DP1-DPk, such as a personal computer and a workstation, are connected through the networks NW, such as a Local Area Network. Moreover, shared memories SM1-SMk are formed in each data processors DP1-DPk, respectively.

[0013] Here, Network NW is equipped with data communication facility, such as a two-way communication function of 1 to 1, and a broadcasting function in which the destination is not specified. Moreover, the node address for specifying on a network is set to each data processors DP1-DPk. Moreover, the exchange of the data between each data-processor DPa-DPk is realized by the server-client model.

[0014] Drawing 2 shows an example of the system configuration of data-processor DP (DP1-DPk). In addition, in this drawing, only the part about the important section of this invention is shown, and explanation is omitted about other parts.

[0015] In this drawing, the task group 1 constitutes the user process performed by this data-processor DP, memory 2 is a memory area used at the time of activation of the task group 1, and the memory management control section 3 is for the task group 1 or the remote memory management control section 4 to manage the actuation which carries out a data access to memory 2.

[0016] The remote memory management control section (RMMS) 4 is a thing because of the data access of shared memory SM (SM1-SMk) formed in memory 2, and is equipped with the server (RMMS/server) function for accessing shared memory SM of a self-node from other nodes, and the client (RMMS/client) function for accessing shared memory SM of other nodes.

[0017] The network control section 5 is for exchanging data among other nodes using the communication facility of Network NW while connecting this data-processor DP to Network NW.

[0018] The field of memory 2 is divided into the shared memory field for constituting shared memory SM, and the local memory field for memorizing the memory object in a self-node etc. as shown in drawing 3.

[0019] In a shared memory field, as shown in drawing 4 (a), while two or more data objects are saved, as shown in this drawing (b), the shared memory management information for accessing each data object is saved. This shared memory management information consists of exclusive mode information which expresses the address where the data object was saved, the size of a data object, the link information which consists of a list of the nodes which link this data object (after-mentioned), the link counter showing the number of the tasks which link this data object, the lock information showing whether this data object is locked, the access restriction to this data object, etc. as shown in this drawing (c).

[0020] The local memory field is divided into the management information field for memorizing further the system area used by the system of a node, and the management information about access of a shared memory, and the user area used by the user task (process) as shown in drawing 5 (a).

[0021] In the management information field, the link control information referred to when accessing the linked shared memory, as shown in this drawing (b), and the shared memory list information which consists of a list of the node addresses holding an effective shared memory are memorized.

[0022] As link control information was shown in this drawing (c), it consists of two or more shared memory reference information, and each shared memory reference information consists of a node address of the node holding the linked shared memory, and reference information (for example, the address and a handle (pointer for specifying the address)) for specifying the shared memory.

[0023] It is the above configuration, for example, a data processor DP<sub>a</sub> creates shared memory 2', and the case where the task of a data processor DP<sub>b</sub> accesses shared memory 2' of this data processor DP<sub>a</sub> is shown in drawing 6. In this case, access-control actuation of shared memory 2' is performed between the server ability of the remote-data supervisory control section 4 of a data processor DP<sub>a</sub>, and the client function of the remote-data supervisory control section 4 of a data processor DP<sub>b</sub>.

[0024] First, shared memory creation task 1a of a data processor DP<sub>a</sub> requires creation of a shared memory by remote memory management control function call, and, thereby, the remote memory management control section 4 secures a physical memory object in the shared memory field of memory 2 as shared memory 2'. And shared memory management information is created and saved based on the real address of the memory object at this time, size, etc. Information (the address and handle) required in order to access the shared memory 2' with it is returned to shared memory creation task 1a.

[0025] With it, the remote memory management control section 4 of a data processor DP<sub>a</sub> notifies the purport which created shared memory 2' to other data-processor DP<sub>b</sub>-DP<sub>k</sub>. Thereby, the remote memory management control section 4 of data-processor DP<sub>b</sub>-DP<sub>k</sub> creates shared memory reference information, and adds it to link control information while it updates shared memory list information about shared memory 2' of this data processor DP<sub>a</sub>.

[0026] Thus, when share memory access task 1b of a data processor DP<sub>b</sub> accesses created shared memory 2', share memory access task 1b opens this shared memory 2' by remote memory-function call.

[0027] Thereby, the remote memory management control section 4 of a data processor DP<sub>b</sub> specifies that the node of specified shared memory 2' is a data processor DP<sub>a</sub> with reference to the contents of the memorized link control information, and requests the link of shared memory 2' to the data processor DP<sub>a</sub>.

[0028] With the data processor DP<sub>a</sub> which received this link request, when the remote memory management control section 4 judges whether the link request at that time is received with reference to the lock information and exclusive mode information on shared memory 2' which were specified and receives a link request, where information (the address and handle) required for a link is set up, it returns having carried out link reception to a data processor DP<sub>b</sub>.

Moreover, the remote memory management control section 4 of a data processor DP<sub>a</sub> increases one value of link count while adding the node address of the data processor DP<sub>b</sub> at that time to the link information of shared memory 2'.

[0029] Since the link was received at this time, the remote memory management control section 4 of a data processor DP<sub>b</sub> notifies link reception to share memory access task 1b, where information required for the notified link is specified.

[0030] Thereby, share memory access task 1b will be in an accessible condition about the shared memory 2'.

[0031] And when share memory access task 1b reads data from shared memory 2', share memory access task 1b is in the condition which added information required in order to specify data to read, such as the address of shared memory 2', a handle, and offset, as a parameter, and performs a remote memory management control function call.

[0032] Thereby, the remote memory management control section 4 of a data processor DP<sub>b</sub> specifies that the node of specified shared memory 2' is a data processor DP<sub>a</sub> with reference to the contents of the memorized link control information, to the data processor DP<sub>a</sub>, is in the condition which specified offset of the read-out address etc., and requests data read-out from shared memory 2'.

[0033] In the data processor DP<sub>a</sub> which received this data read-out request, when the remote memory management control section 4 judges whether the data read-out request at that time is received with reference to the lock information and exclusive mode information on shared memory 2' which were specified and receives a data read-out request, about shared memory 2', data read-out is performed from the offset then specified, and that read-out data is returned to a data processor DP<sub>b</sub>.

[0034] By this, the remote memory management control section 4 of a data processor DP<sub>b</sub>

returns the received data to share memory access task 1b.

[0035] Moreover, when share memory access task 1b writes data in shared memory 2', share memory access task 1b specifies information (the offset by shared memory 2', storing place address, write-in size, etc.) required for writing as a parameter, is in the condition which passed write-in data, and performs a remote memory management control function call.

[0036] Thereby, the remote memory management control section 4 of a data processor DPb is in the condition which passed write-in data, and requests the data writing to shared memory 2' while it specifies that the node of specified shared memory 2' is a data processor DPa with reference to the contents of the memorized link control information and specifies offset etc. to the data processor DPa.

[0037] When the remote memory management control section 4 judges whether the data write-in request at that time is received and receives a data write-in request in the data processor DPa which received this data write-in request with reference to the lock information and exclusive mode information on shared memory 2' which were specified, write-in actuation of the passed write-in data is carried out from the offset then specified about shared memory 2'.

[0038] Moreover, when share memory access task 1b abandons the access privilege of shared memory 2', share memory access task 1b is in the condition which specified shared memory 2' as a parameter, and performs a remote memory management control function call.

[0039] Thereby, the remote memory management control section 4 of a data processor DPb specifies that the node of specified shared memory 2' is a data processor DPa with reference to the contents of the memorized link control information, and notifies the link termination to shared memory 2' to the data processor DPa.

[0040] In the data processor DPa which received this notice of link termination, while the remote memory management control section 4 deletes the node address which corresponds from the link information of shared memory 2', one value of a link counter is reduced. Moreover, since it is the case where there is no process which links that shared memory 2' when the value of a link counter is set to 0 at this time, shared memory 2' is discarded.

[0041] Moreover, when share memory access task 1b wants to perform the lock of shared memory 2', share memory access task 1b is in the condition which specified the range (address range assignment etc.) locked as a parameter, and exclusive mode information, and performs a remote memory management control function call about shared memory 2'.

[0042] Thereby, the remote memory management control section 4 of a data processor DPb specifies that the node of specified shared memory 2' is a data processor DPa with reference to the contents of the memorized link control information, to the data processor DPa, is in the condition which specified the range to lock and exclusive mode information, and notifies the lock request to shared memory 2'.

[0043] In the data processor DPa which received this lock request, when the remote memory management control section 4 judges whether the lock request at that time is received with reference to the lock information and exclusive mode information on shared memory 2' which were specified and receives a lock request, a setup of a lock or exclusive mode information is performed about the range then specified about shared memory 2'.

[0044] Thus, shared memory 2' is created by the data processor DPa, and this shared memory 2' can be accessed from other data processors DPb, and shared memory 2' is discarded.

[0045] Moreover, since shared memory 2' of other data processors DPa can be accessed by performing the same remote memory management control function call as the time of share memory access task 1b of a data processor DPb accessing a local memory, share memory access task 1b can access shared memory 2', without carrying out distinction of a local memory and a shared memory (remote memory).

[0046] Drawing 7 shows an example of the processing which the node which creates a shared memory performs.

[0047] First, creation of a shared memory is requested to the remote memory management control section 4 (processing 101). Thereby, the remote memory management control section 4 creates management information required in order to access it while securing a shared memory (processing 102).

[0048] And if one of tasks requires the data writing to a shared memory (processing 103), to the specified shared memory, data will write the remote memory management control section 4, and it will perform \*\*\*\* (processing 104).

[0049] Drawing 8 shows an example of the processing when reading data from the shared memory of other nodes.

[0050] First, opening of a shared memory is requested to the remote memory management control section 4 (processing 201). Thereby, the remote memory management control section 4 creates the management information for accessing the shared memory while specifying the node holding the specified shared memory (processing 202).

[0051] Next, if read-out of the data from a shared memory is required (processing 203), the remote memory management control section 4 will read data from the shared memory of the purpose node, and will receive the read-out data (processing 204).

[0052] Drawing 9 shows other examples of processing of the creation time of a shared memory.

[0053] First, in the node which creates a shared memory, management information is created (processing 301) and shared memory creation is notified to other nodes (processing 302). Subsequently, to the creation task of a shared memory, reference information is returned (processing 303) and the contents of shared memory list information are updated (processing 304).

[0054] Moreover, in the node which does not create a shared memory, if the notice of shared memory creation is received as shown in drawing 10 (processing 401), the node address of the shared memory creation node at that time will be added to shared memory list information (processing 402).

[0055] Drawing 11 shows the example of processing which the node which requires a link with a shared memory at the time of opening of a shared memory performs.

[0056] First, it investigates whether shared memory list information is then held (decision 501). When the result of decision 501 is set to NO, it asks whether hold the shared memory to each node (processing 502), the response from each node is received (processing 503), and shared memory list information is created based on the receiving result (processing 504).

[0057] When processing 504 is ended, and when the result of decision 501 is set to YES, it shifts to decision 505. In this decision 505, when it investigates whether one or more nodes are registered into shared memory list information and the result of decision 505 is set to YES, to that node, a link request is published (processing 506) and the result of that link request is received (processing 507).

[0058] And when it investigates whether the link was then successful (decision 506) and the result of decision 506 is set to YES, shared memory reference information is created and saved (processing 507).

[0059] Drawing 11 shows the example of processing when receiving an inquiry of shared memory existence.

[0060] Reception of an inquiry of shared memory existence notifies the existence condition of the shared memory at that time to the node of the inquiry origin (processing 602). (processing 601)

[0061] Drawing 12 shows the example of processing when receiving a link request.

[0062] Reception of a link request investigates whether about the request origin, an access privilege etc. can be checked (processing 702) and it can link (decision 703). (processing 701)

[0063] When the result of decision 703 is set to YES, a link information is answered (processing 704) and one value of a link counter is increased (processing 705). A link failure is notified when the result of decision 703 is set to NO (processing 706).

[0064] Drawing 14 shows an example of processing of the node of a requestor side when accessing a shared memory.

[0065] First, shared memory reference information is obtained with reference to the parameter specified from the task (processing 801). The parameters specified at this time are normal values, and when shared memory reference information is obtained correctly, the result of the (decision 802 publishes an access request to the remote memory management control server ability of the node holding YES) and a shared memory (processing 803).

[0066] Next, the response from the remote memory management control server ability is received (processing 804), and it investigates whether access C was then notified (decision 805). When the result of decision 805 is set to YES, it investigates whether data read-out is then performed (decision 806).

[0067] When the result of decision 806 is set to YES, a data read-out demand is published to remote memory management control server ability (processing 808), and the data received by it are passed to the task which carried out the access request (processing 809).

[0068] By the case where data writing is required, when the result of decision 806 is set to NO, a data write request is published to remote memory management control server ability (processing 810), and the data passed from the task are sent out (processing 811).

[0069] Drawing 15 shows an example of processing when an access request is carried out.

[0070] If an access request is received (processing 901), an access privilege etc. will be checked (processing 902) and it will investigate whether it is accessible (decision 903). When the result of decision 903 \*\* is set to NO, an access failure is notified to access request origin (processing 904).

[0071] When the result of decision 903 is set to YES, it investigates whether access C was notified to access request origin (processing 905), and data read-out was then notified (decision 906).

[0072] When the result of decision 906 is set to YES, specified data read-out actuation is performed (processing 907), and the read data is transmitted to access request origin (processing 908).

[0073] Moreover, when the result of decision 906 is set to NO, write-in data are received (processing 909) and it writes in in the mode which specified the received data (processing 910).

[0074] Drawing 16 shows the example of processing when being ordered in link release.

[0075] A link release request is published to the remote memory management control server ability of the node holding the shared memory specified from the task (processing 1001), and the information on the specified shared memory is deleted from shared memory reference information (processing 1002).

[0076] Drawing 17 shows an example of the processing when receiving a link release request.

[0077] If a link release request is received (processing 1101), the node of the demand origin will be eliminated from a link information (processing 1102), and one value of a link counter will be reduced (processing 1103). Here, when it investigates whether the value of a link counter is 0 (decision 1104) and the result of decision 1104 is set to YES, a shared memory and management information are discarded (processing 1105).

[0078] Drawing 18 shows other examples when being ordered in link release.

[0079] If ordered in abandonment of a shared memory from a task (processing 1201), the notice of abandonment of a shared memory will be published to the node then linked to the shared memory (processing 1202), and the response from each node will be received (processing 1203).

[0080] When it investigates whether a shared memory can be discarded (decision 1204) and the result of decision 1204 is set to YES by the case where all the nodes answer abandonment \*\*\*\*, a shared memory and management information are discarded (processing 1205).

[0081] Drawing 19 shows an example of the processing performed when the notice of abandonment of a shared memory is received.

[0082] If the notice of shared memory abandonment is received (processing 1301), when it will investigate whether it is in the condition which can discard the shared memory then notified (decision 1302) and the result of decision 1302 will be set to YES, while notifying abandonment C, shared memory reference information is discarded (processing 1303). Moreover, an abandonment failure is notified when the result of decision 1302 is set to NO (processing 1304).

[0083] By the way, the shared memories SM1-SMk held by each data-processor DP<sub>A</sub>-DP<sub>k</sub> can be arranged in a part of imagination linearity memory.

[0084] In that case, the address where shared memory management information saved the data object as shown in drawing 20 (a), The offset showing the size of a data object, and the start address of the field which this shared memory occupies in an imagination linearity memory area, The link information which consists of a list of the nodes which link this data object, It

constitutes from exclusive mode information showing the link counter showing the number of the tasks which link this data object, the lock information showing whether this data object is locked, the access restriction to this data object, etc.

[0085] Moreover, as shown in this drawing (b), the imaginary line form memory map for memorizing the arrangement situation of the shared memory in virtual linearity memory is formed in each data-processor DP<sub>a</sub>-DP<sub>k</sub>. This imaginary line form memory map consists of two or more share memory maps related with each shared memory. Moreover, the contents of each share memory map serve as offset and a node address from size, as shown in this drawing (c).

[0086] Therefore, two or more shared memories on two or more nodes can be accessed as continuous memory in this case, and the burden of the processing when accessing a shared memory can be mitigated.

[0087] In this case, a task performs a remote memory management control function call, and the example of processing when requiring creation of a shared memory is shown in drawing 21.

[0088] A demand of creation of a shared memory investigates first whether the imaginary line form memory map is then held (decision 1401). When the result of decision 1401 is set to NO, to each node, about the shared memory currently held, the notice of the map information (offset, size, etc.) about a shared memory is required (processing 1402), the notice data from each node are received (processing 1403), and an imaginary line form memory map is created based on the received data at that time (processing 1404).

[0089] When processing 1404 is ended, and when the result of decision 1401 is set to YES, it investigates whether a shared memory is securable for a real storage (decision 1405).

[0090] When the result of decision 1405 is set to YES, based on an imaginary line form memory map, the offset value of the shared memory to create is determined in imaginary line form memory, and the management information of the shared memory is created (processing 1406). At this time, the next location of the block located in the most high-order address can be considered as offset, or a shared memory can be secured to a free area.

[0091] And the purport which created the shared memory, and the information about a shared memory are notified to each node about the created shared memory (processing 1407).

[0092] Drawing 22 shows an example of the processing performed when the notice demand of map information of a shared memory is received.

[0093] Reception of the notice demand of map information of a shared memory transmits then the map information on the shared memory currently held by the self-node to the demand node (processing 1502). (processing 1501)

[0094] Drawing 23 shows an example of the processing performed when the notice of shared memory creation is received.

[0095] Reception of the notice of shared memory creation updates the contents of the imaginary line form memory map from the contents then notified (processing 1602). (processing 1601)

[0096] A task performs a remote memory management control function call, and the example of processing when requiring access to a shared memory is shown in drawing 24. At this time, information which specifies a shared memory, such as offset of a shared memory which carries out an access request, is specified as a parameter.

[0097] Reception of the access request to a shared memory investigates first whether the imaginary line form memory map is then held (decision 1701). When the result of decision 1701 is set to NO, to each node, about the shared memory currently held, the notice of the map information (offset, size, etc.) about a shared memory is required (processing 1702), the notice data from each node are received (processing 1703), and an imaginary line form memory map is created based on the received data at that time (processing 1704).

[0098] When processing 1704 is ended, and when the result of decision 1701 is set to YES, it investigates whether a shared memory exists in the offset then specified (decision 1705).

[0099] When the result of decision 1705 is set to YES, a link is required from the node holding a shared memory (processing 1706), and the response from the node is received (processing 1707).

[0100] When it investigates whether the link was successful (decision 1708) and the result of decision 1708 is set to YES at this time, shared memory reference information is created

(processing 1709).

[0101] Drawing 25 shows the example of processing when receiving a link demand.

[0102] First, it investigates whether about the link demand origin at that time, an access privilege can be checked (processing 1801) and it can link (decision 1802). When the result of decision 1802 is set to YES, the link information about a shared memory is created and it sends out to link demand origin (processing 1803). Moreover, when the result of decision 1802 is set to NO, a link failure is notified to link demand origin (processing 1804).

[0103] By the way, if the identifier of a meaning is added to a shared memory, since the target thing can be specified among two or more shared memories on a network only by specifying this identifier, assignment of a shared memory becomes easy.

[0104] In that case, shared memory management information consists of exclusive mode information showing the address where the data object was saved as shown in drawing 26 (a), the size of a data object, the identifier added to this shared memory, the link information which consists of a list of the nodes which link this data object, the link counter showing the number of the tasks which link this data object, the lock information showing whether this data object is locked, the access restriction to this data object, etc.

[0105] Moreover, as shown in this drawing (b), the shared memory identifier table showing the relation between a shared memory and an identifier is formed in each data-processor DP<sub>A</sub>-DP<sub>K</sub>. This shared memory identifier table consists of shared memory identifier information about each shared memory. Moreover, the contents of each shared memory identifier information serve as an identifier from a node address, as shown in this drawing (c).

[0106] In this case, the example of processing when requiring creation of a shared memory, where the identifier which a task performs a remote memory management control function call, and adds is specified is shown in drawing 27.

[0107] A demand of creation of a shared memory investigates first whether the shared memory identifier table is then held (decision 1901). When the result of decision 1901 is set to NO, to each node, about the shared memory currently held, the identifier of a shared memory is asked (processing 1902), the notice data from each node are received (processing 1903), and a shared memory identifier table is created based on the received data at that time.

[0108] When the result of decision 1901 is set to YES after ending processing 1903 and, the identifier then specified investigates whether it is already added to other shared memories (decision 1904).

[0109] When it investigates whether a shared memory can newly be then created when the result of decision 1904 is set to YES (decision 1905) and the result of decision 1905 is set to YES, the purport which created the shared memory, and the information about a shared memory are notified to each node about the created shared memory (processing 1906).

[0110] Drawing 28 shows an example of the processing performed when an identifier inquiry of a shared memory is received.

[0111] Reception of an identifier inquiry of a shared memory transmits then the identifier of the shared memory currently held by the self-node to the demand node (processing 2002). (processing 2001)

[0112] Drawing 29 shows an example of the processing performed when the notice of shared memory creation is received.

[0113] Reception of the notice of shared memory creation updates the contents of the shared memory identifier table from the contents then notified (processing 2102). (processing 2101)

[0114] A task performs a remote memory management control function call, and the example of processing when requiring access to a shared memory is shown in drawing 30. At this time, the information which specifies shared memories, such as an identifier of the shared memory which carries out an access request, is specified as a parameter.

[0115] Reception of the access request to a shared memory investigates first whether the shared memory identifier table is then held (decision 2201). When the result of decision 2201 is set to NO, to each node, about the shared memory currently held, the identifier of a shared memory is asked (processing 2202), the notice data from each node are received (processing 2203), and a shared memory identifier table is created based on the received data at that time.

[0116] When the result of decision 2201 is set to YES after ending processing 2203 and, the identifier then specified investigates whether it is contained in the shared memory identifier table (decision 2204).

[0117] When the result of decision 2204 is set to YES, a link is required from the node holding a shared memory (processing 2205), and the response from the node is received (processing 2206).

[0118] When it investigates whether the link was successful (decision 2207) and the result of decision 2207 is set to YES at this time, shared memory reference information is created (processing 2208).

[0119] Now, this invention is applicable similarly about the case where the node equipped with the virtual-memory-management device is used.

[0120] Drawing 31 shows an example of the system configuration of the data processor equipped with the virtual-memory-management device. In addition, in this drawing, only the part about the important section of this invention is shown, and explanation is omitted about other parts. Moreover, in this drawing, the same sign is given to the same part as drawing 2, and the corresponding part.

[0121] In this drawing, task group 1' constitutes the user process performed by this data-processor DP, memory 2 is a memory area used at the time of activation of the task group 1, and memory management control-section 3' is for the virtual-memory-management control section (virtual-memory-management device) 6 or remote memory management control-section 4' to manage the actuation which carries out a data access to memory 2.

[0122] Remote memory management control-section 4' is a thing because of the data access of shared memory SM (SM1-SM<sub>k</sub>) formed in memory 2, and is equipped with the server (RMMS/server) function for accessing shared memory SM of a self-node from other nodes, and the client (RMMS/client) function for accessing shared memory SM of other nodes.

[0123] The virtual-memory-management control section 6 is for realizing a virtual-memory-management device, and task group 1' accesses memory 2 through this virtual-memory-management control section 6.

[0124] In this case, in order that memory 2 may carry out a data access, a descriptor as shown in drawing 32 (a) is used. This descriptor consists of memory information for accessing each memory object.

[0125] Moreover, the address which shows the physical address of the storage region where this memory information corresponds to this drawing (b) as shown, Remote one / local information for distinguishing whether they are size and remote memory or it is a local memory, The swap information showing whether swapping out of the data is carried out to secondary storage, The link information which consists of a list of the nodes which link this data object, It consists of exclusive mode information showing the link counter showing the number of the tasks which link this data object, the lock information showing whether this data object is locked, the access restriction to this data object, etc.

[0126] In this case, a task performs a remote memory management control function call, and the example of processing when requiring the link to a shared memory is shown in drawing 33.

[0127] If a task performs a remote memory management control function call and requires the link to a shared memory (processing 2301), since the virtual-address information for share memory access (memory information) will return from a remote memory management controlling mechanism (processing 2302), a task saves this virtual-address information.

[0128] Drawing 34 is a remote memory management control function call, and shows the example of processing by the side of remote memory management control-section 4' when the link to a shared memory is required, and a system.

[0129] First, remote memory management control-section 4' creates a link information (processing 2401), and creates the descriptor (memory information) with which a remote state corresponds (processing 2402). And the virtual-address information corresponding to the descriptor is returned to a task (processing 2403).

[0130] Moreover, a task performs a remote memory management control function call, and the example of processing when requiring access to a shared memory is shown in drawing 35.

[0131] First, a task requires access of a shared memory from a remote memory management controlling mechanism using the virtual-address information then returned from the system (remote memory management controlling mechanism) (processing 2501).

[0132] Thereby, if access C is notified from a system (the result of decision 2502 is YES), proper data processing will be then performed to the shared memory which carried out the access request (processing 2503).

[0133] Drawing 36 shows an example of the processing which a system performs, when a memory access demand is carried out from a task.

[0134] First, a corresponding descriptor (memory information) is obtained from the virtual address specified from the task (processing 2601). And it investigates whether the candidate for access is remote memory (decision 2602).

[0135] When the result of decision 2602 is set to NO, the access privilege about the task is checked (processing 2603), and it investigates whether access is good (decision 2604). When the result of decision 2604 is set to NO, an access failure is notified to the task which carried out the access request (processing 2605).

[0136] Moreover, when it investigates whether swapping out of the memory object at that time is carried out when the result of decision 2604 is set to YES (decision 2606) and the result of decision 2606 is set to YES, the data is loaded on a real storage (processing 2607).

[0137] Thus, if it will be in the condition in which a data access is possible, data-access processing will be performed to a task (processing 2608).

[0138] Moreover, the object by which the access request was carried out is remote memory (shared memory), when the result of decision 2602 is set to YES, based on the descriptor then obtained, the link information for accessing a shared memory is formed (processing 2609), and an access request is published to the remote memory management control server of the node holding the shared memory (processing 2610).

[0139] To the access request at that time, when access C is notified, the result of the (decision 2611 shifts to YES) and processing 2608, and data-access processing is performed to a task. Moreover, in the time of an access failure being notified, when the result of decision 2611 is set to NO, it shifts to processing 2605 and an access failure is notified to a task.

[0140] Therefore, at this example, by using a virtual-memory-management device, since a task can be performed without distinguishing access to a shared memory from access to a local memory, the burden of processing by the side of a task mitigates it more.

[0141] In addition, this invention is applicable similarly about things other than a system which was mentioned above.

[0142]

[Effect of the Invention] Since memory is sharable between different nodes according to this invention as explained above, sharing of the data in the whole network system is easily realizable. Moreover, since direct access of the shared memory space can be carried out using a virtual-memory-management device when it has the virtual-memory-management device, the effectiveness that the access efficiency of data is good and can access a shared memory at a high speed is acquired.

---

[Translation done.]

**\* NOTICES \***

JPO and INPI are not responsible for any damages caused by the use of this translation.

- 1.This document has been translated by computer. So the translation may not reflect the original precisely.
- 2.\*\*\*\* shows the word which can not be translated.
- 3.In the drawings, any words are not translated.

---

**DESCRIPTION OF DRAWINGS**

---

**[Brief Description of the Drawings]**

[Drawing 1] The block diagram having shown the local-area network system concerning one example of this invention.

[Drawing 2] The block diagram having shown an example of a data processor (node).

[Drawing 3] The schematic diagram having shown an example of division of a memory area.

[Drawing 4] The schematic diagram having shown an example of the data memorized to a shared memory field.

[Drawing 5] The schematic diagram having shown an example of the data memorized to a local memory field.

[Drawing 6] The model block diagram for explaining various processing actuation of access to a shared memory etc.

[Drawing 7] The flow chart which showed the example of processing of the node which forms a shared memory.

[Drawing 8] The flow chart which showed the example of processing of other nodes which do not form a shared memory.

[Drawing 9] The flow chart which showed the example of processing at the time of shared memory formation.

[Drawing 10] The flow chart which showed the example of processing when receiving the notice of shared memory creation.

[Drawing 11] The flow chart which showed the example of processing when carrying out a link request to the shared memory.

[Drawing 12] The flow chart which showed an example of the processing when receiving a shared memory existence inquiry.

[Drawing 13] The flow chart which showed an example of the processing when receiving a link request.

[Drawing 14] The flow chart which showed the example of processing when carrying out an access request to a shared memory.

[Drawing 15] The flow chart which showed the example of processing when receiving an access request.

[Drawing 16] The flow chart which showed the example of processing when requiring link release of a shared memory.

[Drawing 17] The flow chart which showed the example of processing when receiving the notice of link release.

[Drawing 18] The flow chart which showed the example of processing of the abandonment demand of a shared memory.

[Drawing 19] The flow chart which showed the example of processing when receiving the notice of shared memory abandonment.

[Drawing 20] The schematic diagram having shown examples, such as management information concerning other examples of this invention.

[Drawing 21] The flow chart which showed other examples of processing of the creation demand of a shared memory.

[Drawing 22] The flow chart which showed an example of the processing when receiving the notice demand of a share memory map.

[Drawing 23] The flow chart which showed the example of processing when receiving shared memory information.

[Drawing 24] The flow chart which showed other examples of the processing when carrying out a link request to the shared memory.

[Drawing 25] The flow chart which showed the example of processing when receiving a link request.

[Drawing 26] The schematic diagram having shown examples, such as management information concerning the example of further others of this invention.

[Drawing 27] The flow chart which showed the example of processing of further others of the creation demand of a shared memory.

[Drawing 28] The flow chart which showed the example of processing when receiving an inquiry of a shared memory identifier.

[Drawing 29] The flow chart which showed the example of processing when receiving shared memory information.

[Drawing 30] The flow chart which showed the example of processing of further others at the time of a link request.

[Drawing 31] The block diagram having shown the data processor concerning another example of this invention.

[Drawing 32] The schematic diagram having shown an example of the descriptor of the equipment of drawing 31.

[Drawing 33] The flow chart which showed the example of processing of the task at the time of a shared memory open request.

[Drawing 34] The flow chart which showed the example of processing of the system at the time of a shared memory open request.

[Drawing 35] The flow chart which showed the example of processing of the task at the time of share memory access.

[Drawing 36] The flow chart which showed the example of processing of the system at the time of share memory access.

[Description of Notations]

NW Network

DP1- DPk and DP Data processor

SM1- SMk and SM Shared memory

---

[Translation done.]